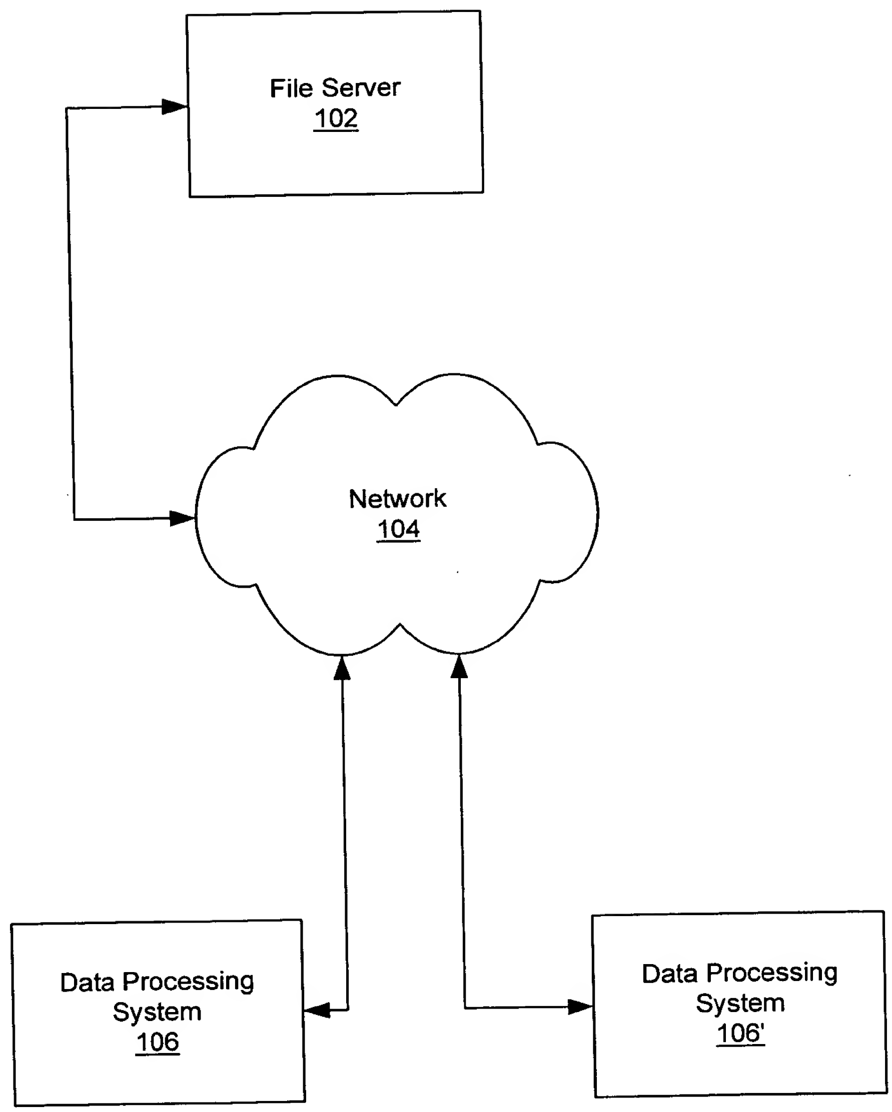
File Server
102
Network
104
Data Processing
System
106
Data Processing
System
106'

Figure 1

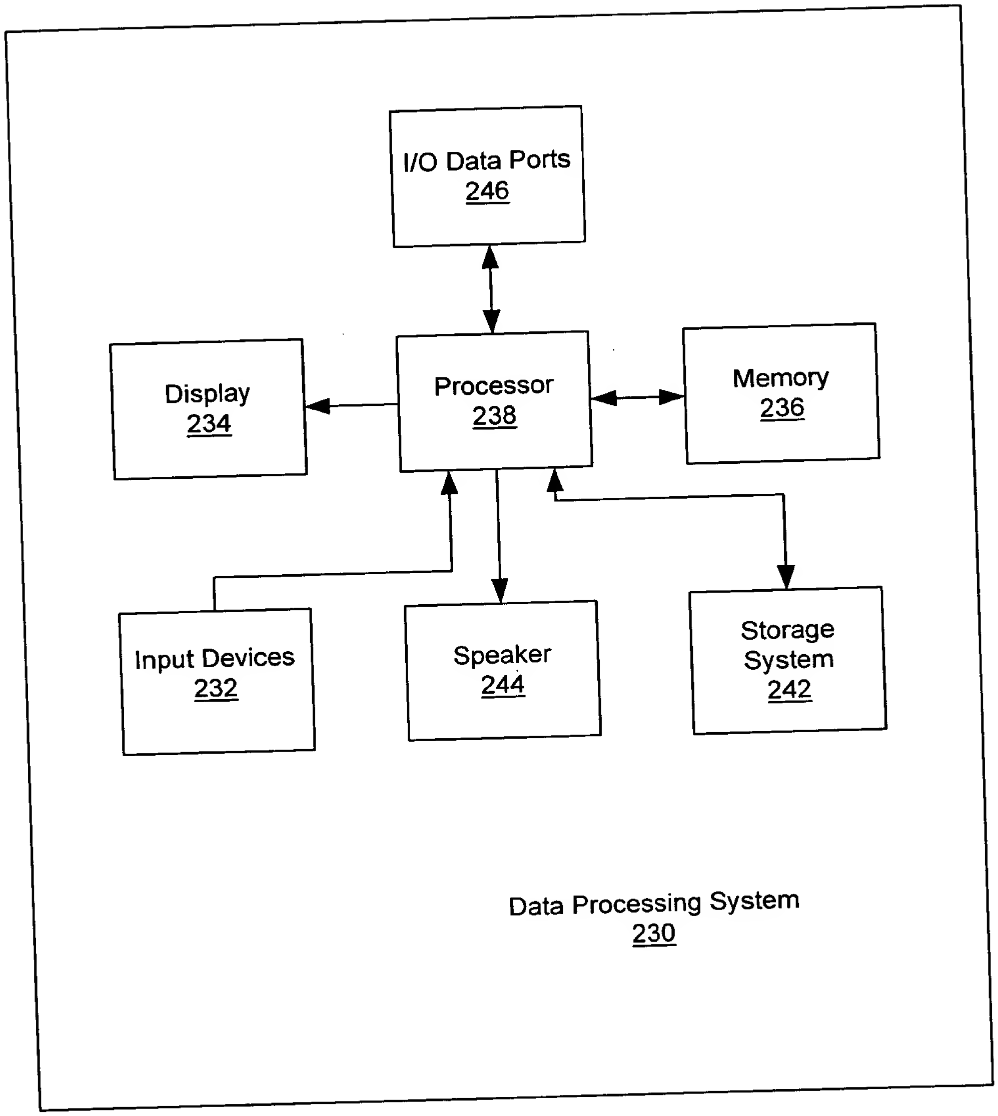

Figure 2

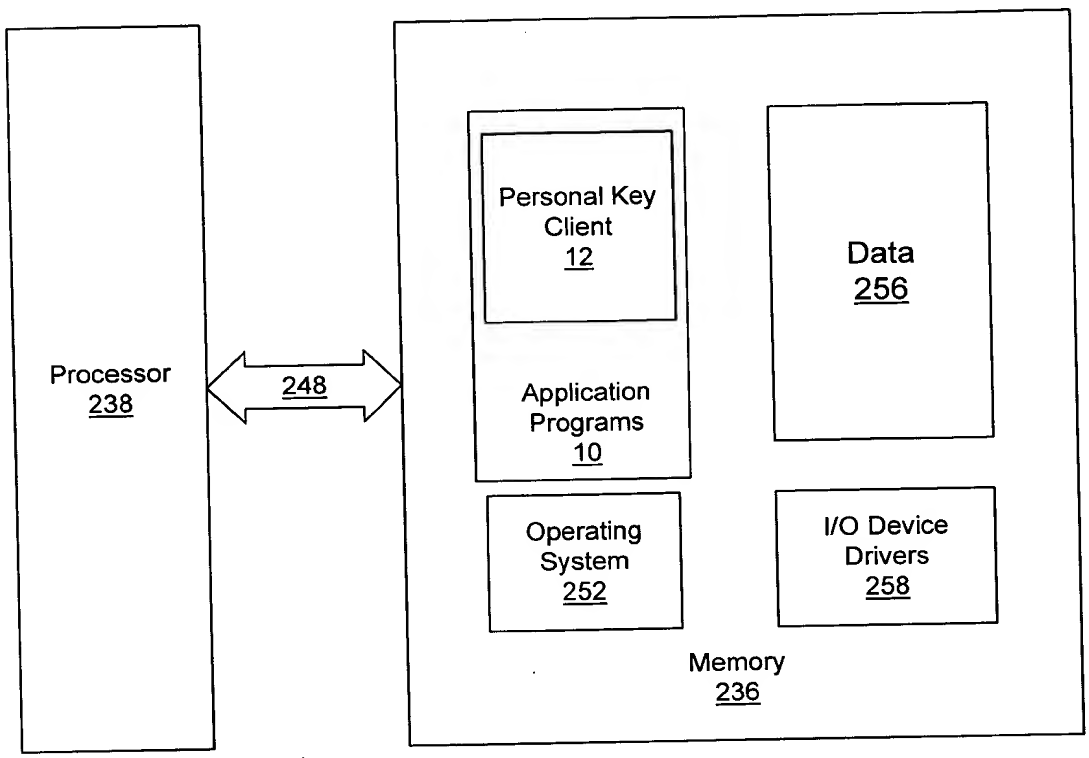
Processor
238
248
Personal Key
Client
12
Application
Programs
10
Data
256
Operating
System
252
I/O Device
Drivers
258
Memory
236

Figure 3

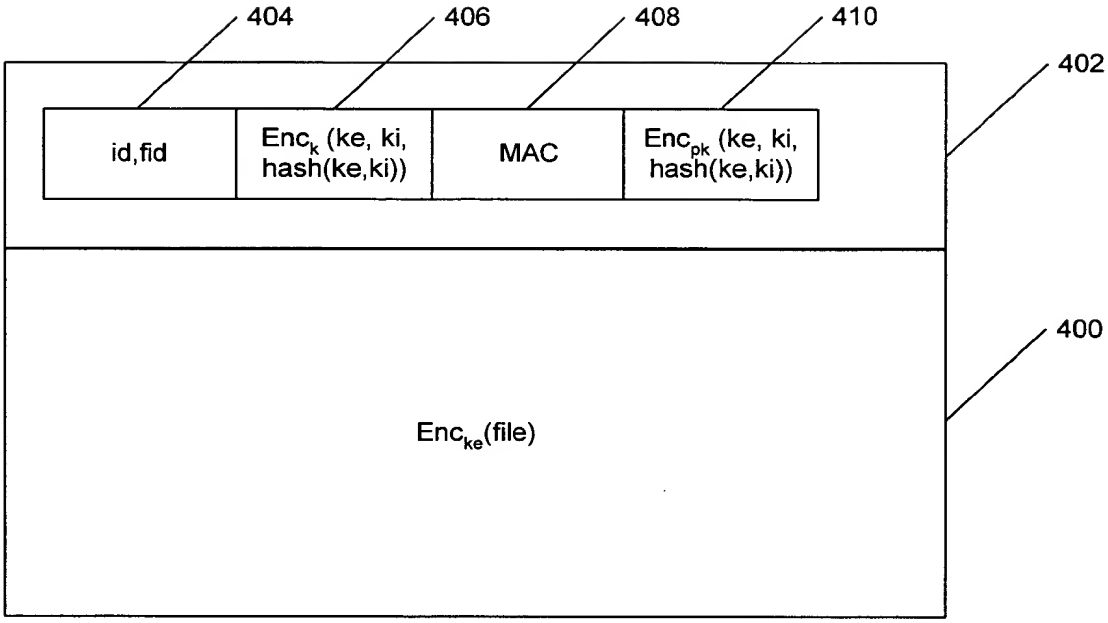
404
406
408
410
402
id,fid
Enc$_k$ (ke, ki, hash(ke,ki))
MAC
Enc$_{pk}$ (ke, ki, hash(ke,ki))
400
Enc$_{ke}$(file)

Figure 4

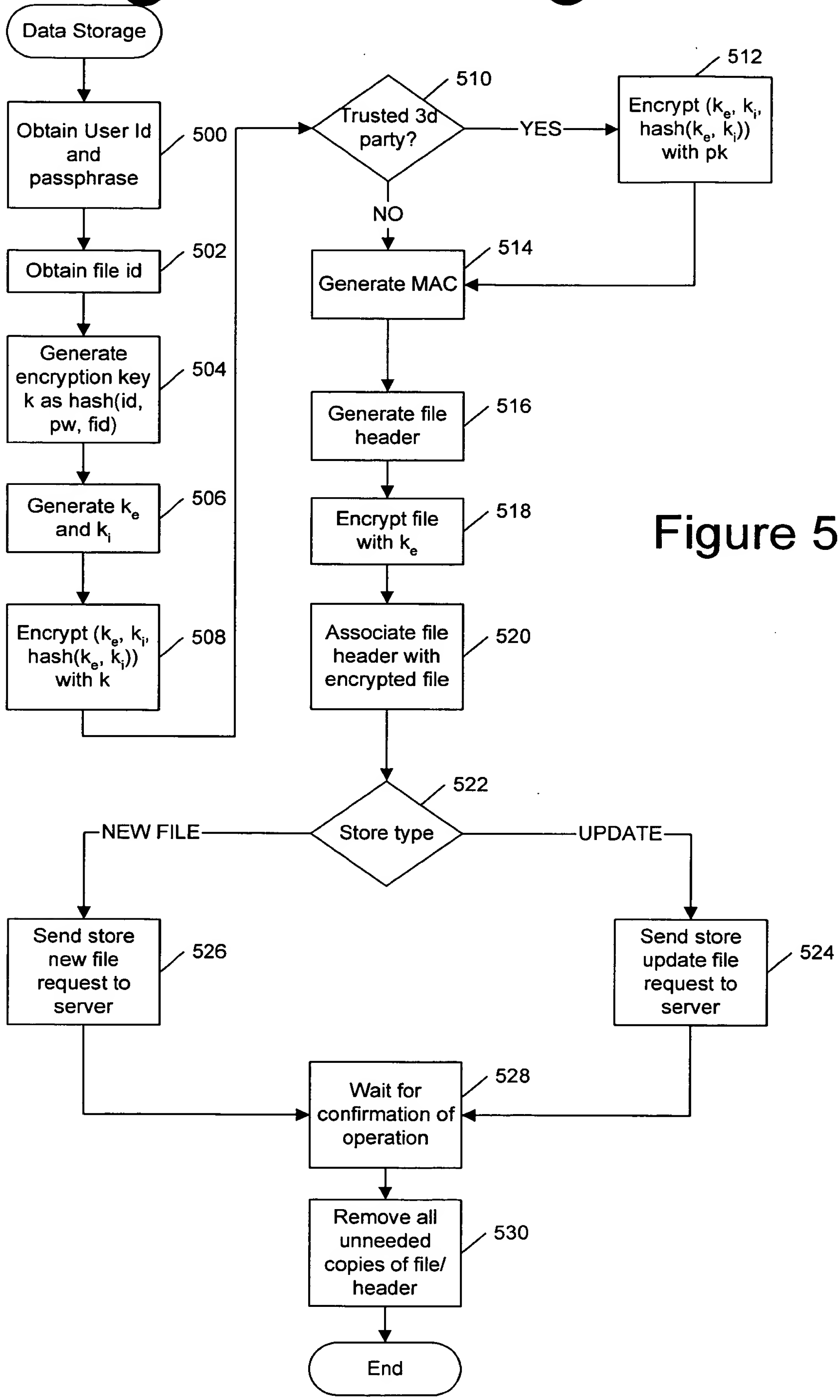

Figure 5

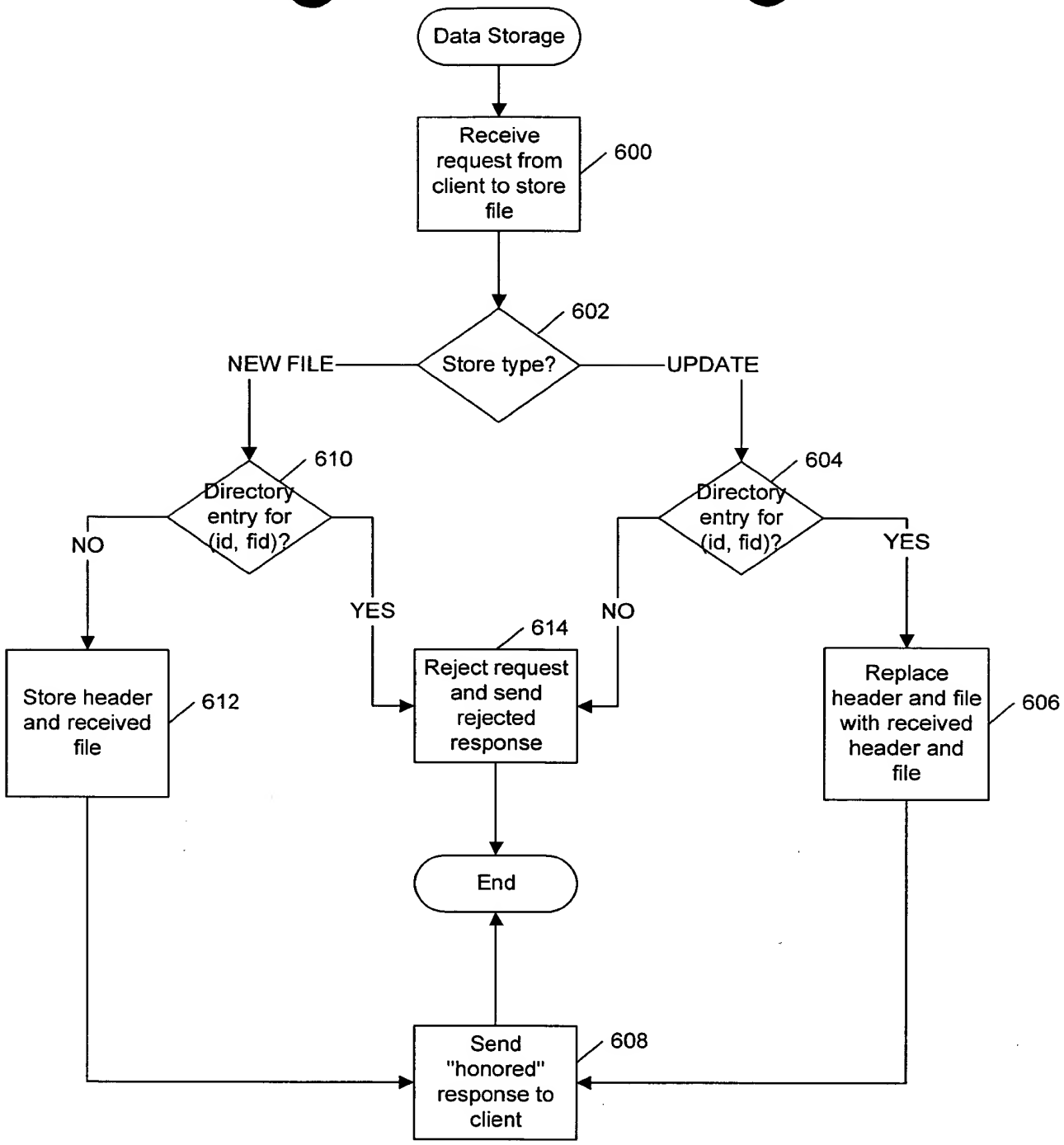
Data Storage
Receive request from client to store file
600
602
Store type?
NEW FILE
UPDATE
610
Directory entry for (id, fid)?
NO
YES
604
Directory entry for (id, fid)?
NO
YES
614
Reject request and send rejected response
Store header and received file
612
Replace header and file with received header and file
606
End
Send "honored" response to client
608

Figure 6

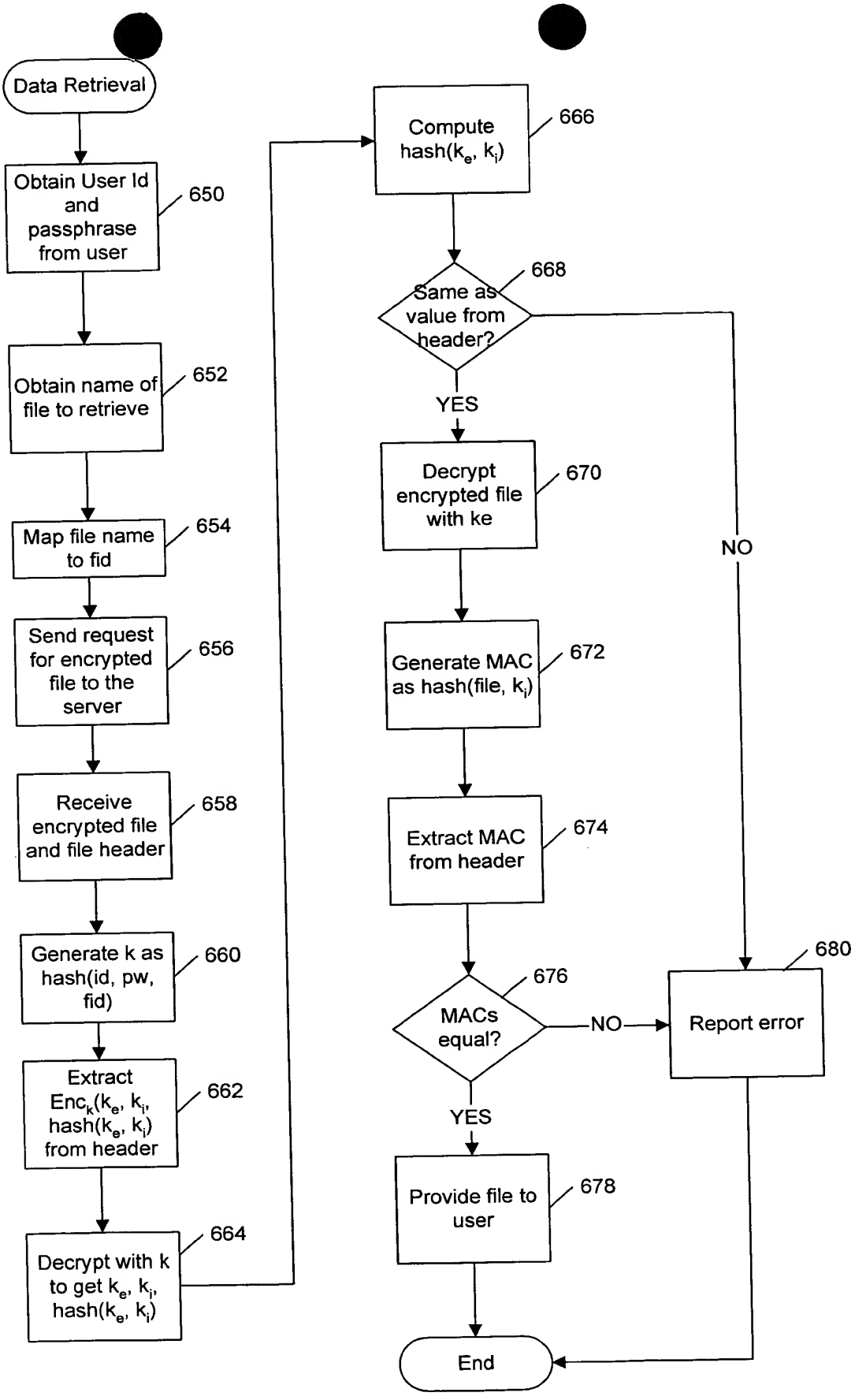
Data Retrieval
Obtain User Id and passphrase from user
650
Obtain name of file to retrieve
652
Map file name to fid
654
Send request for encrypted file to the server
656
Receive encrypted file and file header
658
Generate k as hash(id, pw, fid)
660
Extract $Enc_k(k_e, k_i, hash(k_e, k_i)$ from header
662
Decrypt with k to get k_e, k_i, $hash(k_e, k_i)$
664
Compute $hash(k_e, k_i)$
666
Same as value from header?
668
YES
NO
Decrypt encrypted file with ke
670
Generate MAC as $hash(file, k_i)$
672
Extract MAC from header
674
MACs equal?
676
NO
Report error
680
YES
Provide file to user
678
End

Figure 7

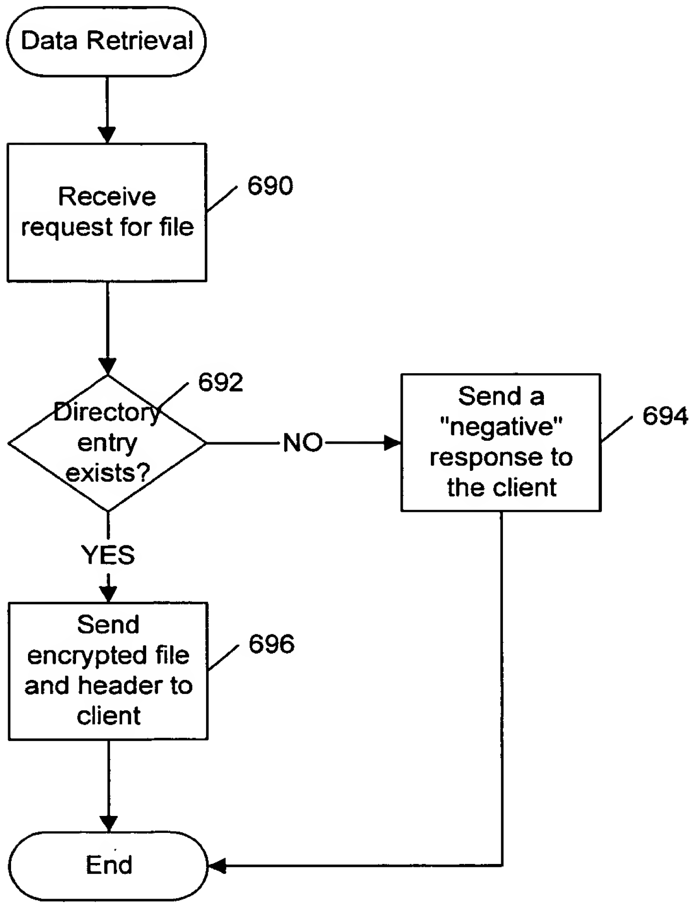
Data Retrieval
Receive request for file
690
Directory entry exists?
692
NO
Send a "negative" response to the client
694
YES
Send encrypted file and header to client
696
End

Figure 8

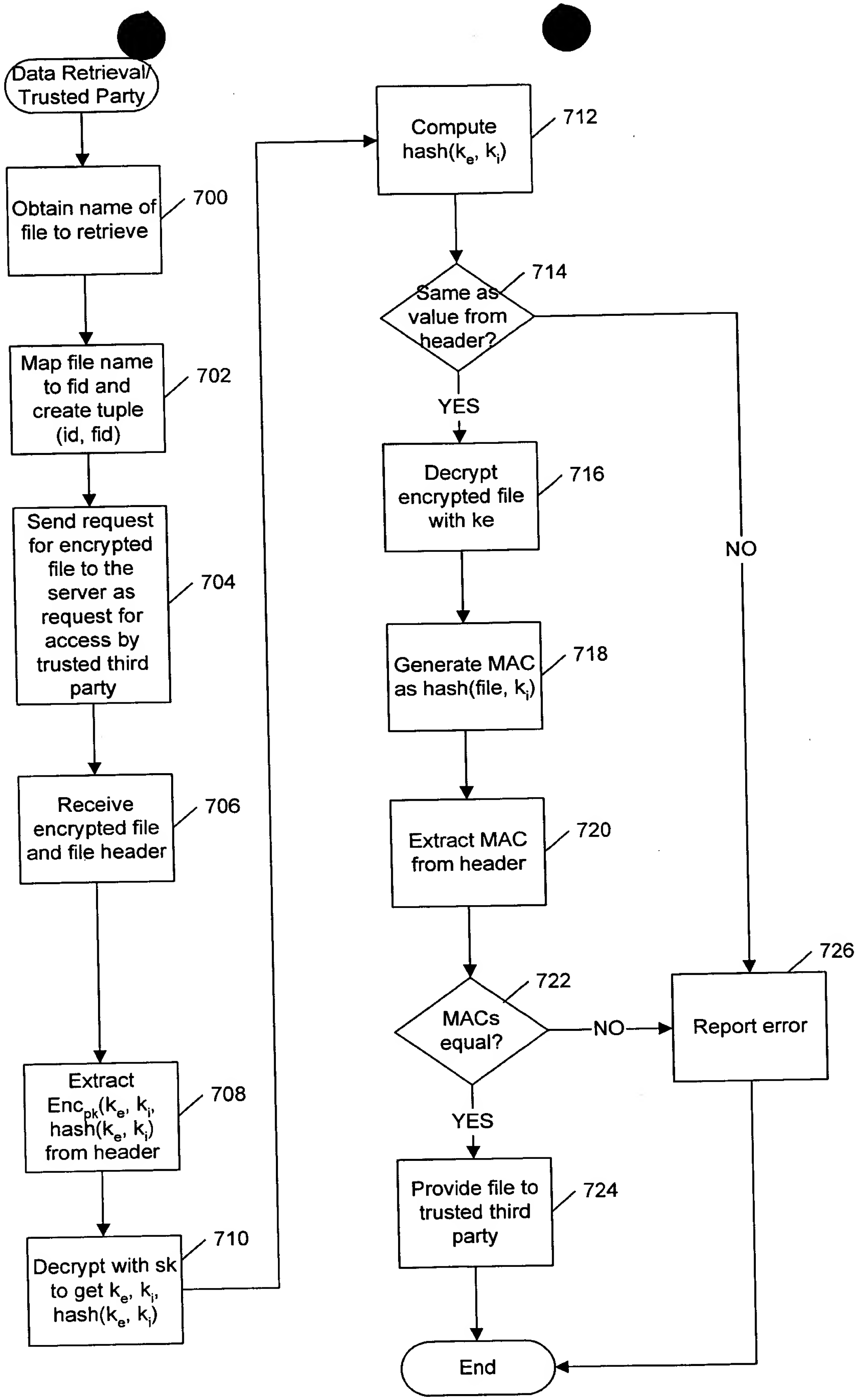
Data Retrieval/ Trusted Party
Obtain name of file to retrieve
700
Map file name to fid and create tuple (id, fid)
702
Send request for encrypted file to the server as request for access by trusted third party
704
Receive encrypted file and file header
706
Extract $Enc_{pk}(k_e, k_i, hash(k_e, k_i))$ from header
708
Decrypt with sk to get k_e, k_i, $hash(k_e, k_i)$
710
Compute $hash(k_e, k_i)$
712
Same as value from header?
714
YES
NO
Decrypt encrypted file with ke
716
Generate MAC as $hash(file, k_i)$
718
Extract MAC from header
720
MACs equal?
722
YES
NO
Provide file to trusted third party
724
Report error
726
End

Figure 9

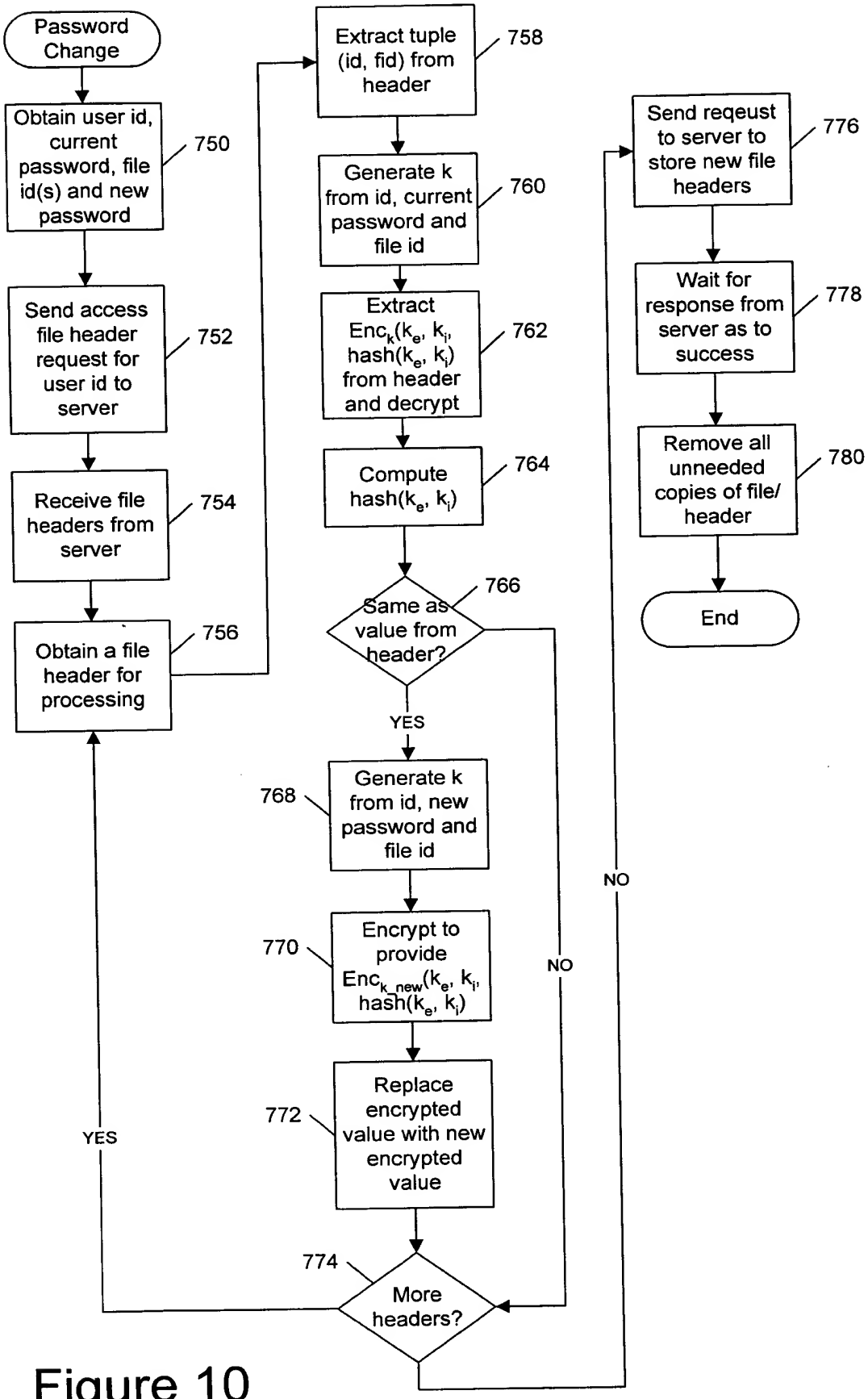
Password Change
Obtain user id, current password, file id(s) and new password
750
Send access file header request for user id to server
752
Receive file headers from server
754
Obtain a file header for processing
756
Extract tuple (id, fid) from header
758
Generate k from id, current password and file id
760
Extract $Enc_k(k_e, k_i)$, $hash(k_e, k_i)$ from header and decrypt
762
Compute $hash(k_e, k_i)$
764
Same as value from header?
766
YES
NO
Generate k from id, new password and file id
768
Encrypt to provide $Enc_{k\_new}(k_e, k_i)$, $hash(k_e, k_i)$
770
Replace encrypted value with new encrypted value
772
More headers?
774
YES
NO
Send reqeust to server to store new file headers
776
Wait for response from server as to success
778
Remove all unneeded copies of file/ header
780
End

Figure 10

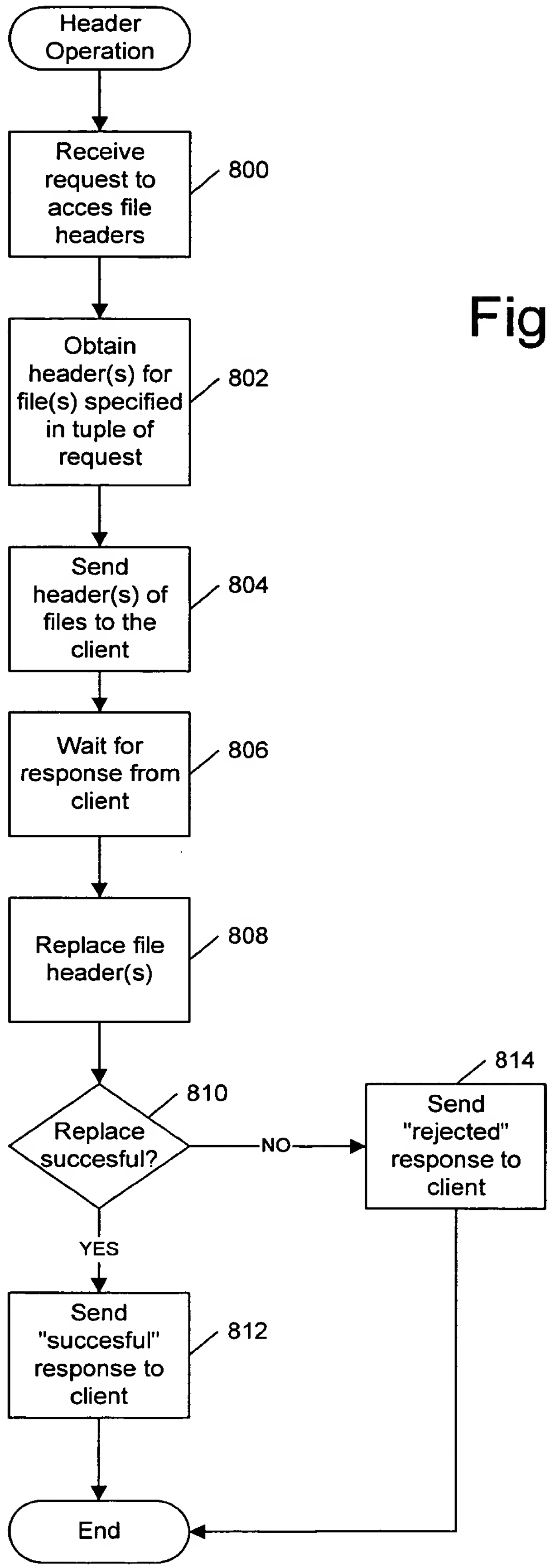

Figure 11

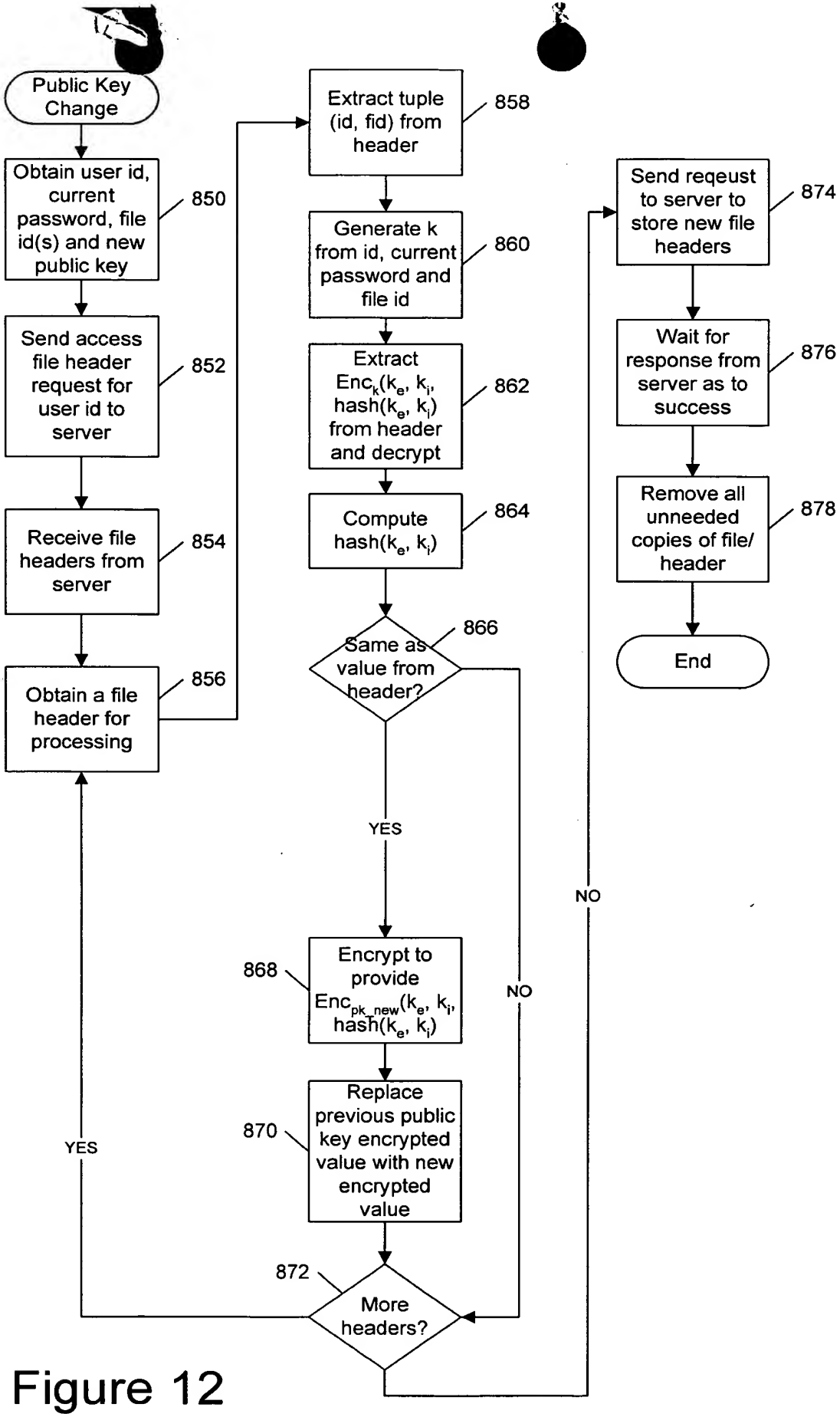
Public Key Change
Obtain user id, current password, file id(s) and new public key
850
Send access file header request for user id to server
852
Receive file headers from server
854
Obtain a file header for processing
856
Extract tuple (id, fid) from header
858
Generate k from id, current password and file id
860
Extract $Enc_k(k_e, k_i)$, $hash(k_e, k_i)$ from header and decrypt
862
Compute $hash(k_e, k_i)$
864
Same as value from header?
866
YES
NO
Encrypt to provide $Enc_{pk\_new}(k_e, k_i)$, $hash(k_e, k_i)$
868
Replace previous public key encrypted value with new encrypted value
870
More headers?
872
YES
NO
Send reqeust to server to store new file headers
874
Wait for response from server as to success
876
Remove all unneeded copies of file/ header
878
End

Figure 12